

NextIM开发指南

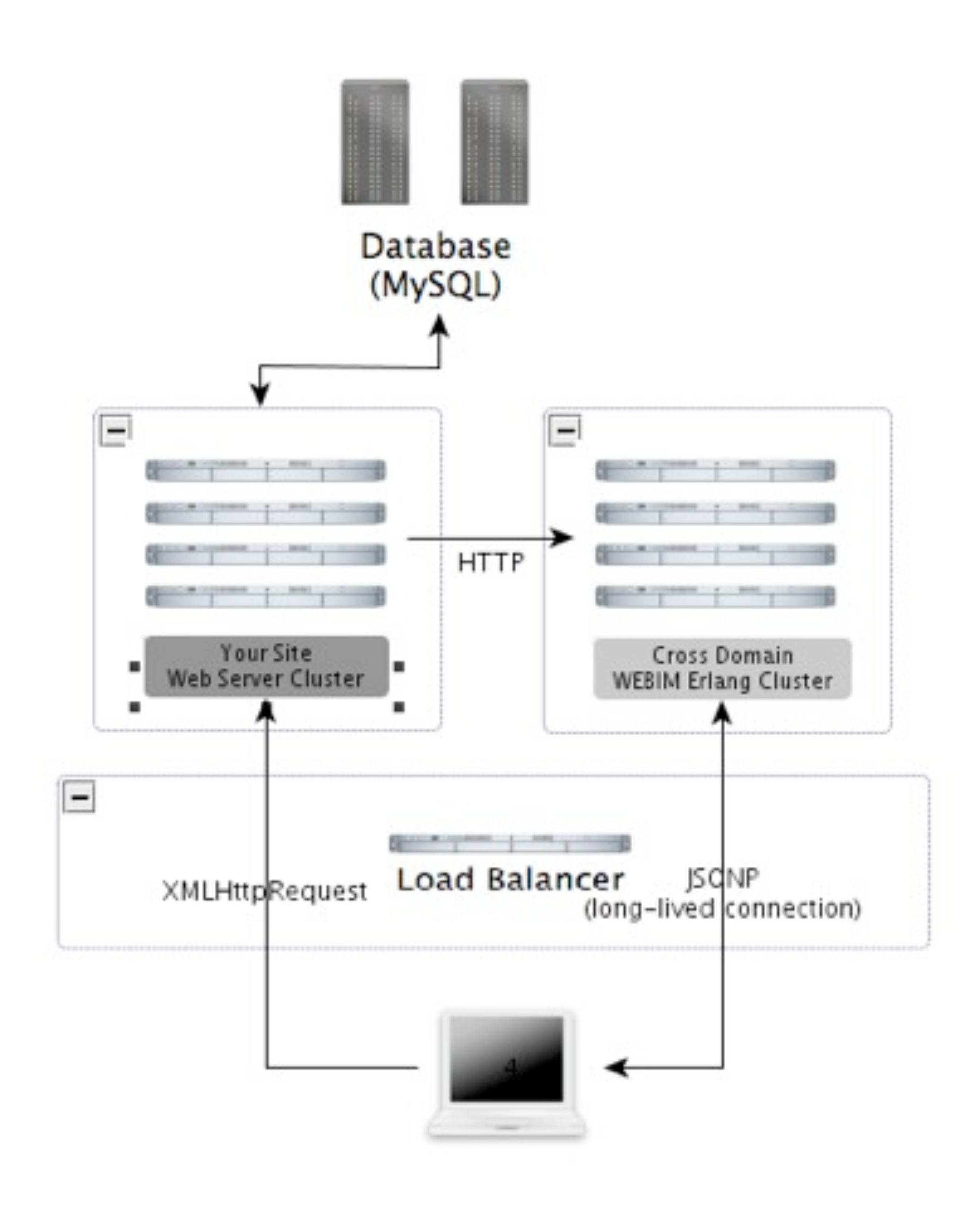
杭州巨鼎信息技术有限公司

目录

NextIM技术架构

总体架构

NextIM是业界领先的基于Erlang、JSONP等技术构建的WebIM服务器，后台架构与GTalk in Gmail、Facebook IM相似，通过JSONP实现跨域长轮询，极大减轻WEBIM对社区站点的性能压力，通过Erlang语言实现可大规模分布式集群的WebIM服务器，可轻松支持100万并发用户。



模块接口

完整的NextIM系统可分为三个独立模块，包括：

- 浏览器的Javascipt模块：该模块独立于任何类型的社区网站，通过AJAX与社区服务器交互，通过JSONP长连接轮训消息服务器。
- 社区服务器器的嵌入WEBIM插件：该插件用于NextIM与社区用户认证、好友关系的集成，完全由社区编写，支持任何类型的编程语言，例如PHP、Java、.Net、Python等。
- 消息路由服务器：消息服务器独立部署，由Erlang编写，主要处理消息路由和大量的JSONP并发长连接。

三个模块或系统间的接口，包括：

- 用户浏览器->社区服务器: AJAX接口，具体见下面章节介绍；
- 社区服务器->消息服务器: HTTP接口，REST风格，主要目的是转发用户上下线信息和路由消息；
- 用户浏览器->消息服务器: JSONP接口，用户通过JSONP长连接进行消息轮训。

技术优势

1. 基于Erlang技术的大规模并发集群服务器

业界领先的基于Erlang、JSONP等技术构建的WebIM服务器，后台架构与Facebook IM相似，通过Erlang语言实现大规模分布式集群的WebIM服务器，可轻松支持100万并发用户。

2. 基于JSONP长轮训技术的准实时即时消息

通过JSONP长轮询技术，在WEB上模拟桌面IM系统的长连接，实现快速、实时的消息发送接收。

3. 采用变革性的SaaS服务模式，使用简单方便

无需安装独立的WebIM服务器，只要申请在线服务即可，租用空间的社区站点也可使用。

4. 支持无中断页面切换和多页面消息同步等WEBIM的核心技术!

WEBIM与桌面IM软件有不同的应用场景，比如用户频繁切换页面，同时打开多个网站页面等。基于创新性的系统架构设计和长期的WEBIM设计经验，我们拥有解决这些问题的核心技术。

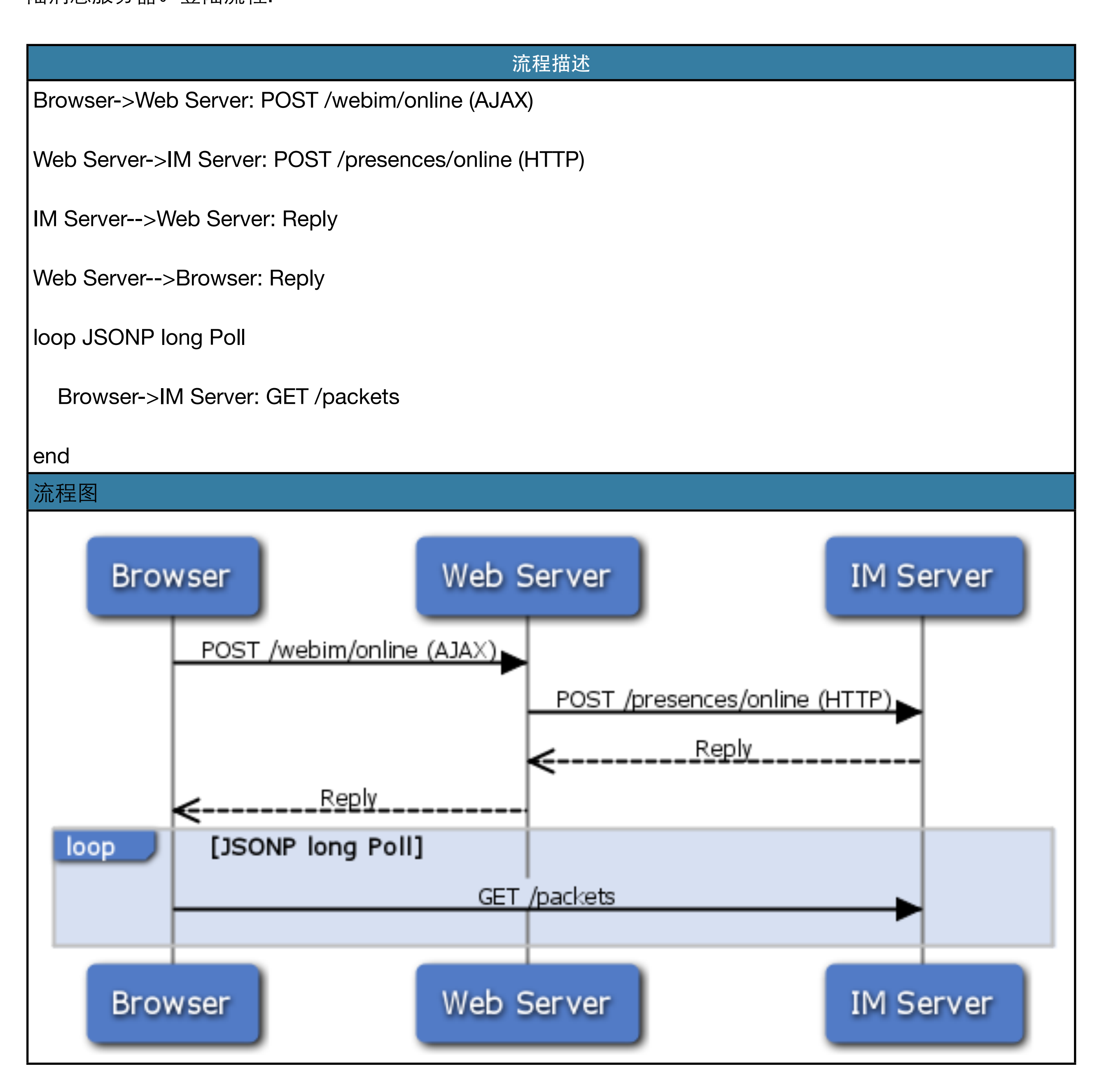
5. 对社区站点几乎无任何性能压力

基于我们创新性的JSONP跨域长轮询技术，有效的避免了WebIM对社区主站本身造成的性能压力。单纯依赖PHP实现的WebIM产品(例如AJAX IM)会通过频繁的轮询来检查即时消息，往往对SNS主站造成非常大的性能影响。

NextIM业务流程

用户上线，登陆消息服务器流程

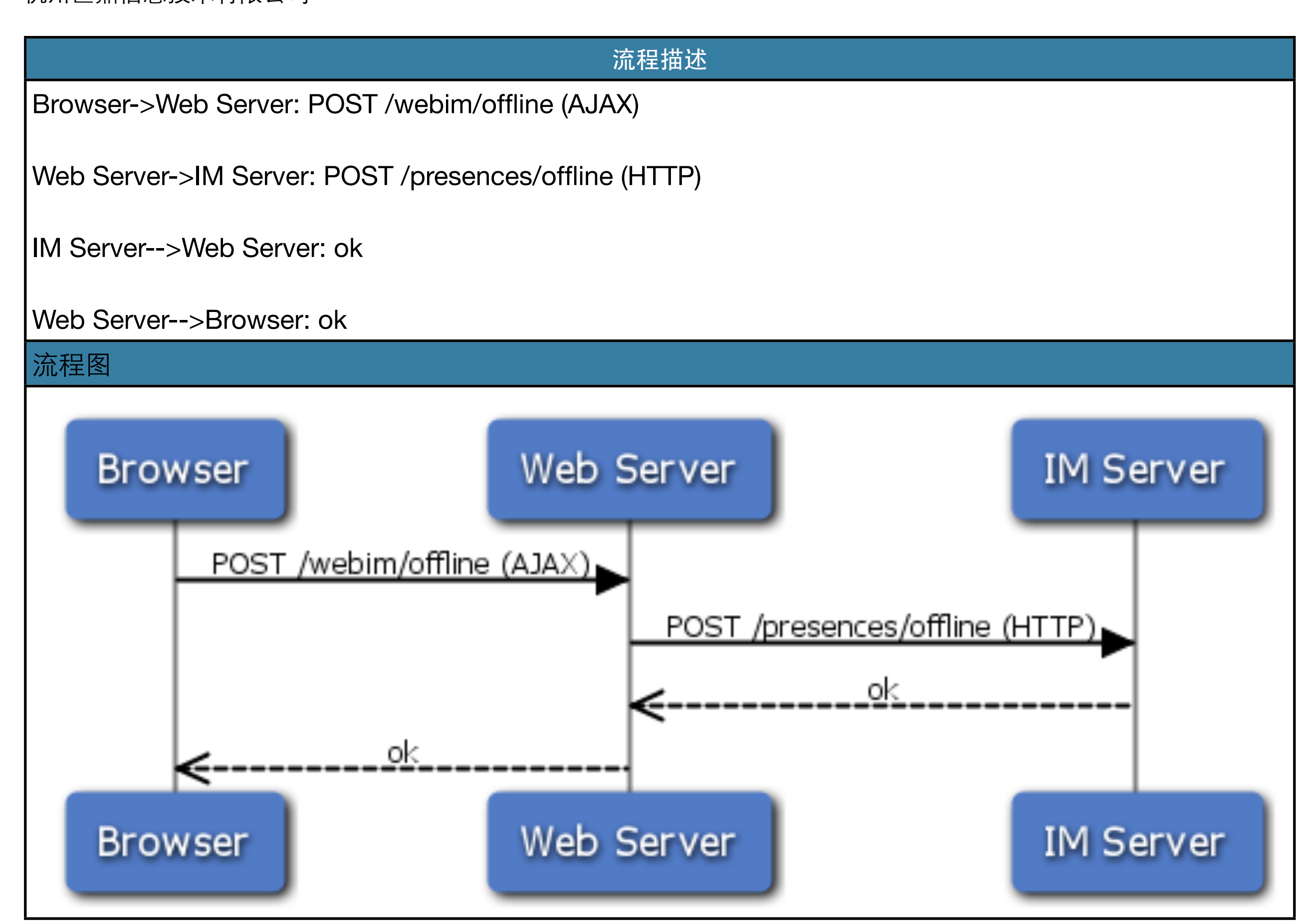
用户打开社区网站页面，WebIM插件相关的Javascript等资源文件加载完毕，界面出现后，会发送AJAX请求登陆消息服务器。登陆流程:



流程描述
Browser->Web Server: POST /webim/online (AJAX) Web Server->IM Server: POST /presences/online (HTTP) IM Server-->Web Server: Reply Web Server-->Browser: Reply loop JSONP long Poll   Browser->IM Server: GET /packets end
流程图

用户离线、登出消息服务器流程

用户退出社区网站、或直接关闭社区网站页面、或手工设置“离线“，将触发用户离线、登出消息服务器流程。



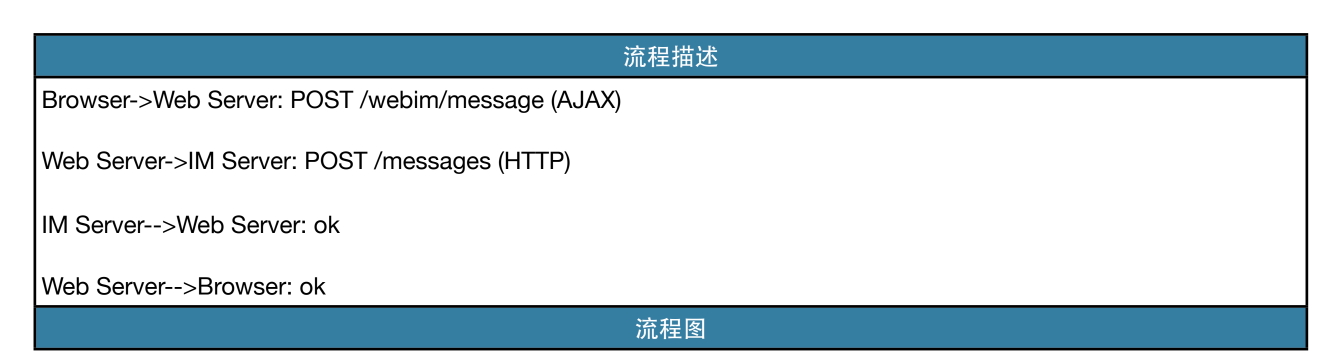
流程描述
Browser->Web Server: POST /webim/offline (AJAX) Web Server->IM Server: POST /presences/offline (HTTP) IM Server-->Web Server: ok Web Server-->Browser: ok
流程图

用户发送即时消息流程

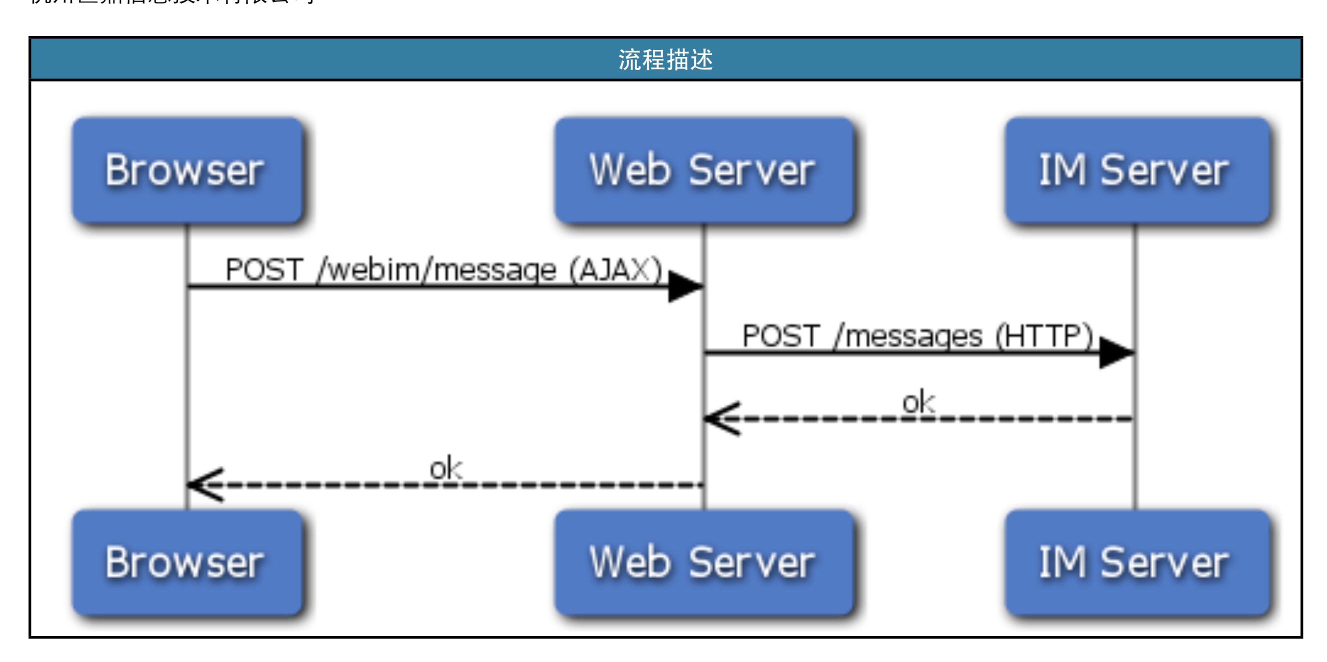
用户登陆消息服务器成功后，可在“联系人”中选取好友打开聊天窗口，发送即时消息，界面如下：



消息发送流程:



流程描述
Browser->Web Server: POST /webim/message (AJAX) Web Server->IM Server: POST /messages (HTTP) IM Server-->Web Server: ok Web Server-->Browser: ok
流程图



前台自动检测并发送输入状态流程

用户打开聊天窗口与好友通信时，前台界面会自动检测用户的输入状态，并向对方发送。状态消息发送流程：

流程描述
Browser->Web Server: POST /webim/status (AJAX) Web Server->IM Server: POST /statuses (HTTP) IM Server-->Web Server: ok Web Server-->Browser: ok
流程图
Browser / Web Server / IM Server POST /webim/status (AJAX) POST /statuses (HTTP) ok ok Browser / Web Server / IM Server

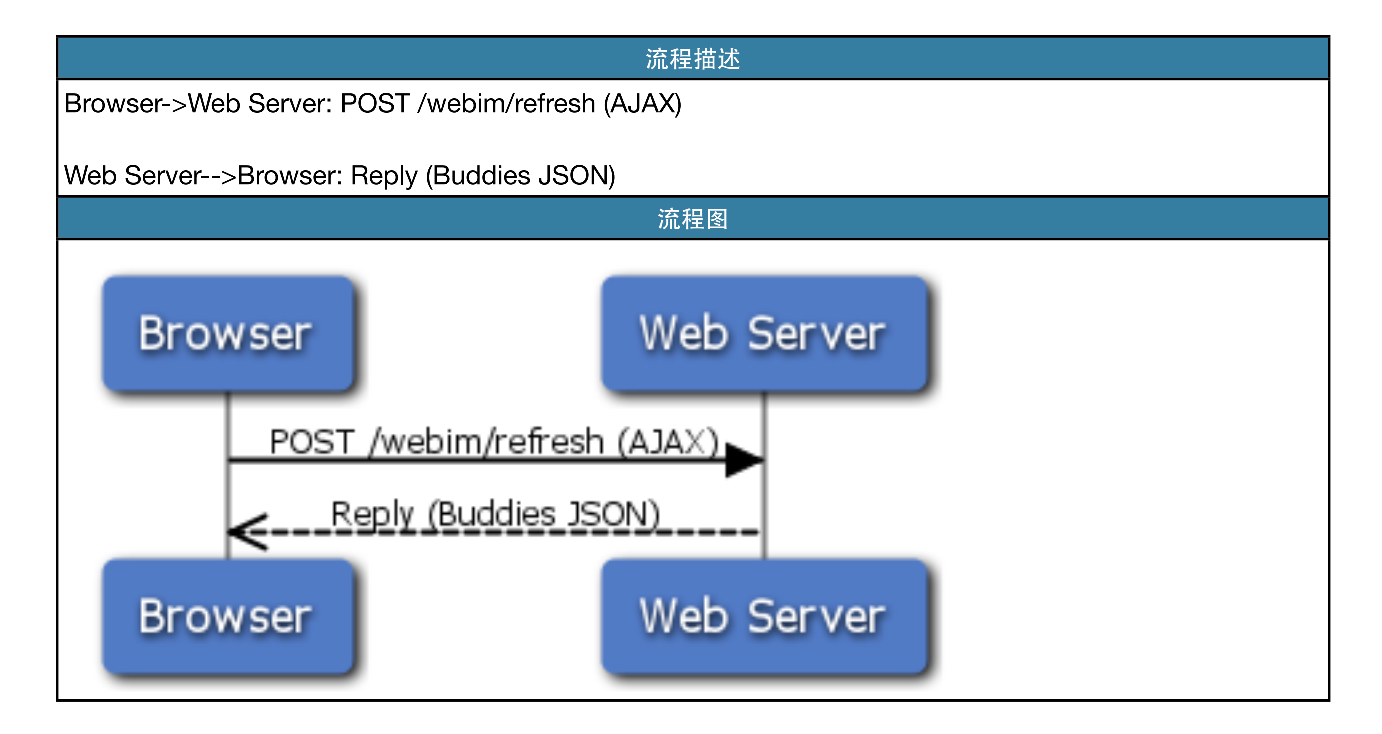
用户刷新或切换社区页面

用户刷新或切换当前浏览的社区页面。触发业务流程如下：

流程描述
Browser->Web Server: POST /webim/refresh (AJAX) Web Server->IM Server: POST /presences/offline (HTTP) IM Server-->Web Server: ok Web Server-->Browser: ok
流程图
Browser Web Server IM Server POST /webim/refresh (AJAX) POST /presences/offline (HTTP) ok ok Browser Web Server IM Server

前台获取好友信息流程

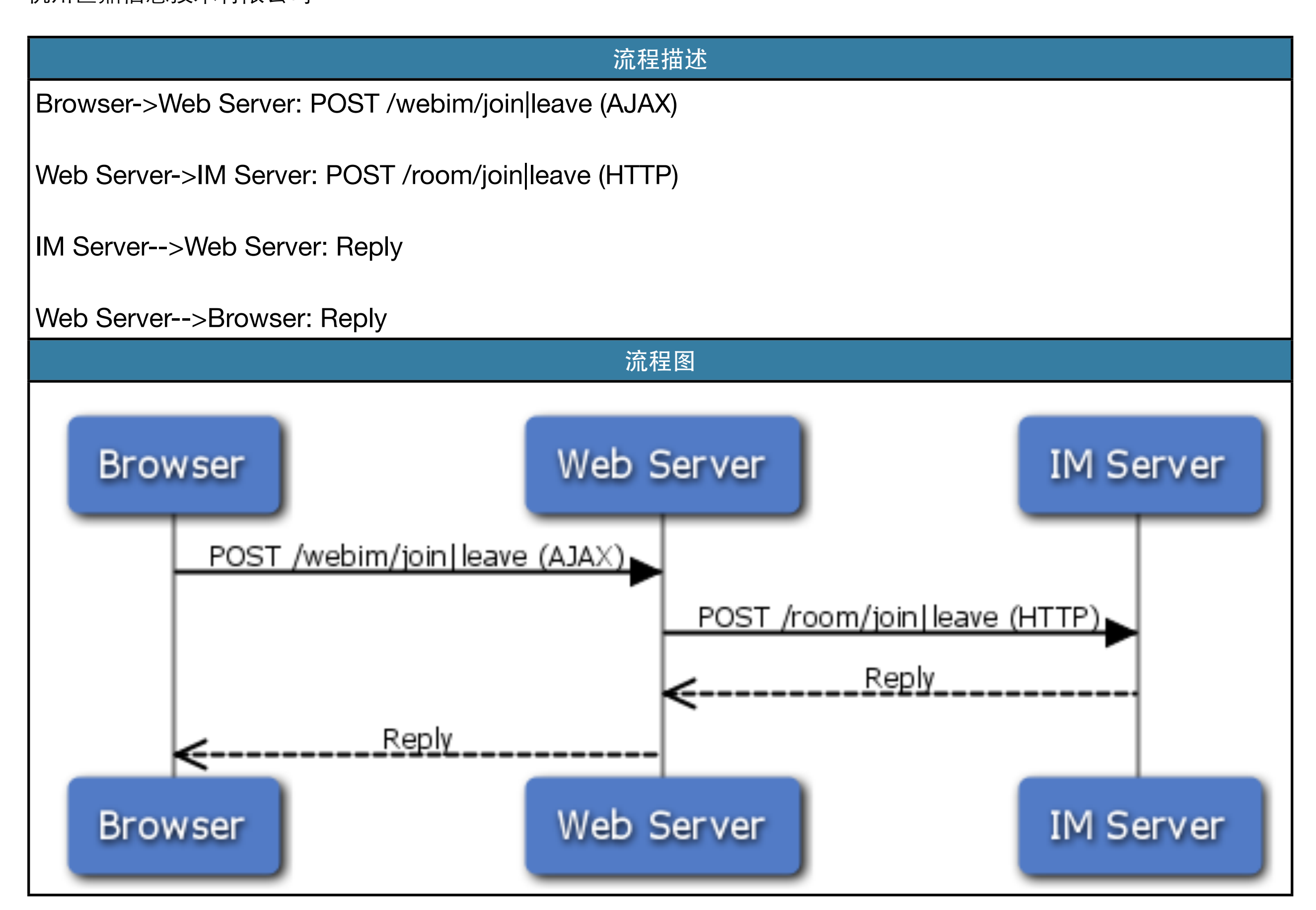
前台收到好友上线的presence现场消息后， 触发读取好友详细信息的业务流程：



流程描述
Browser->Web Server: POST /webim/refresh (AJAX) Web Server-->Browser: Reply (Buddies JSON)
流程图

用户加入离开房间(Room)流程

用户打开群组聊天窗口，可以通过解锁、加锁按钮，触发加入、离开房间流程：



流程描述
Browser->Web Server: POST /webim/join|leave (AJAX) Web Server->IM Server: POST /room/join|leave (HTTP) IM Server-->Web Server: Reply Web Server-->Browser: Reply
流程图

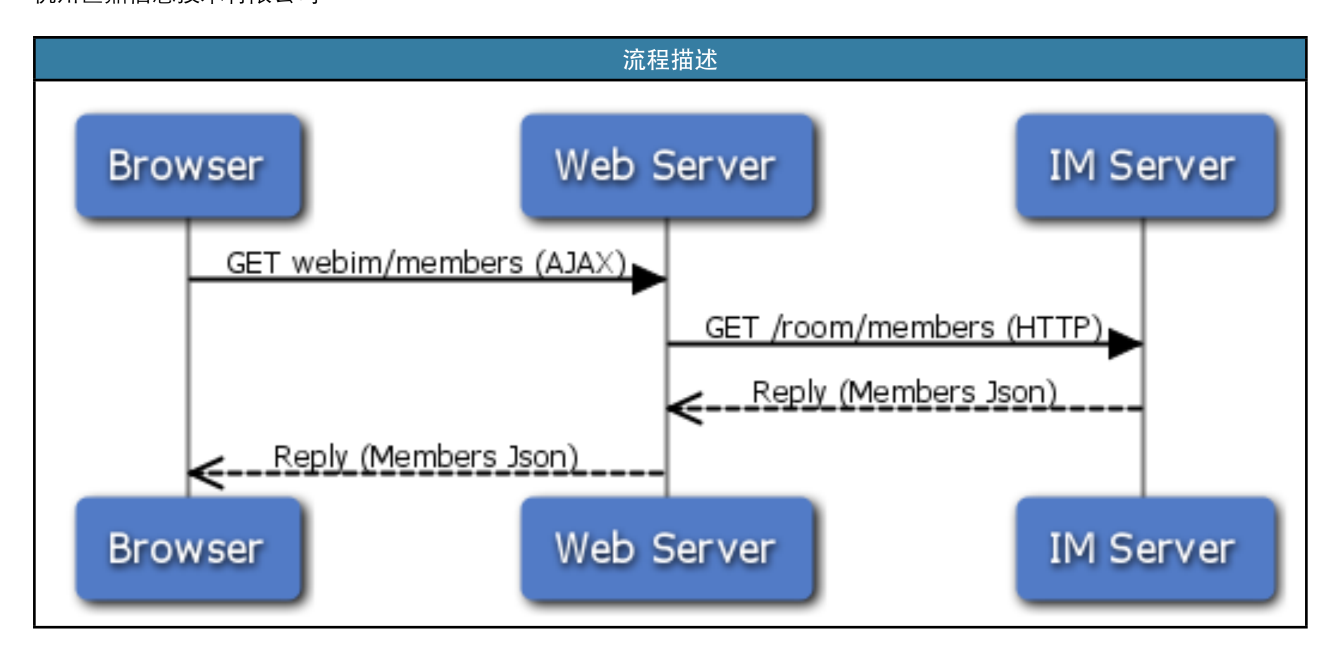
获取房间在线用户列表流程

用户打开房间群聊窗口后，会触发获取房间在线用户列表流程，房间在线用户会显示界面如下：



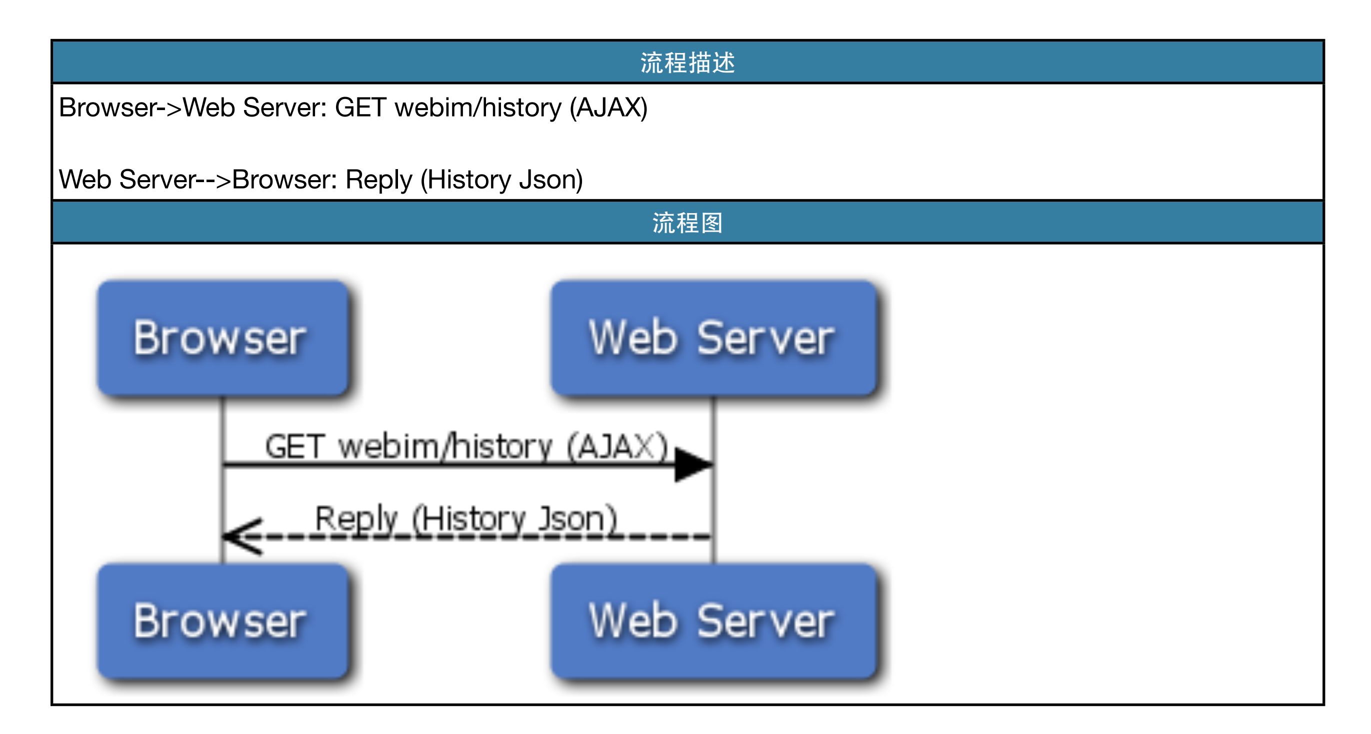
流程如下：

流程描述
Browser->Web Server: GET webim/members (AJAX) Web Server->IM Server: GET /room/members (HTTP) IM Server-->Web Server: Reply (Members Json) Web Server-->Browser: Reply (Members Json)
流程图



获取聊天历史记录流程

用户打开聊天窗口，前台会自动加载双方的历史聊天记录，流程如下：

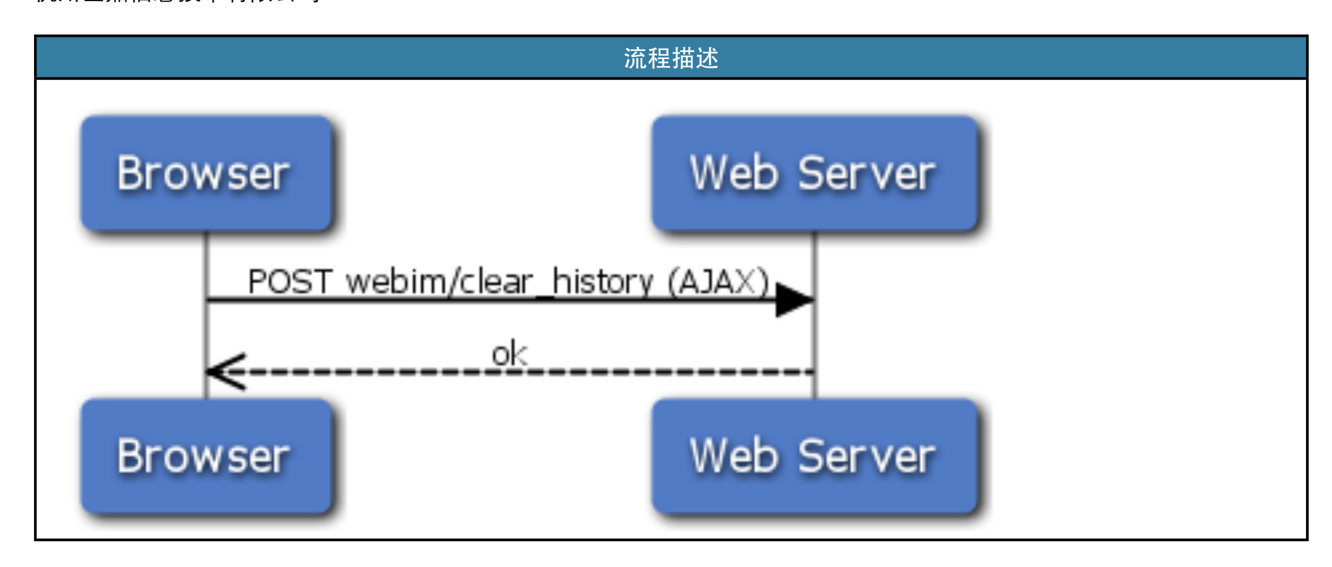


流程描述
Browser->Web Server: GET webim/history (AJAX) Web Server-->Browser: Reply (History Json)
流程图

清除聊天历史记录流程

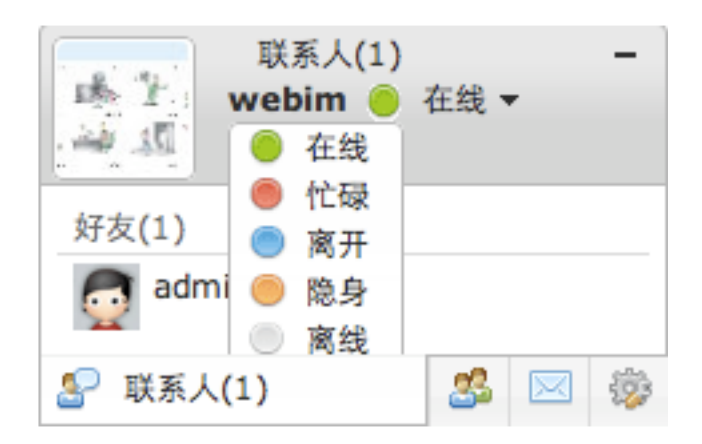
用户打开聊天窗口，可以通过”清除“按钮，清除界面上的聊天记录，流程如下：

流程描述
Browser->Web Server: POST webim/clear_history (AJAX) Web Server-->Browser: ok
流程图

流程描述


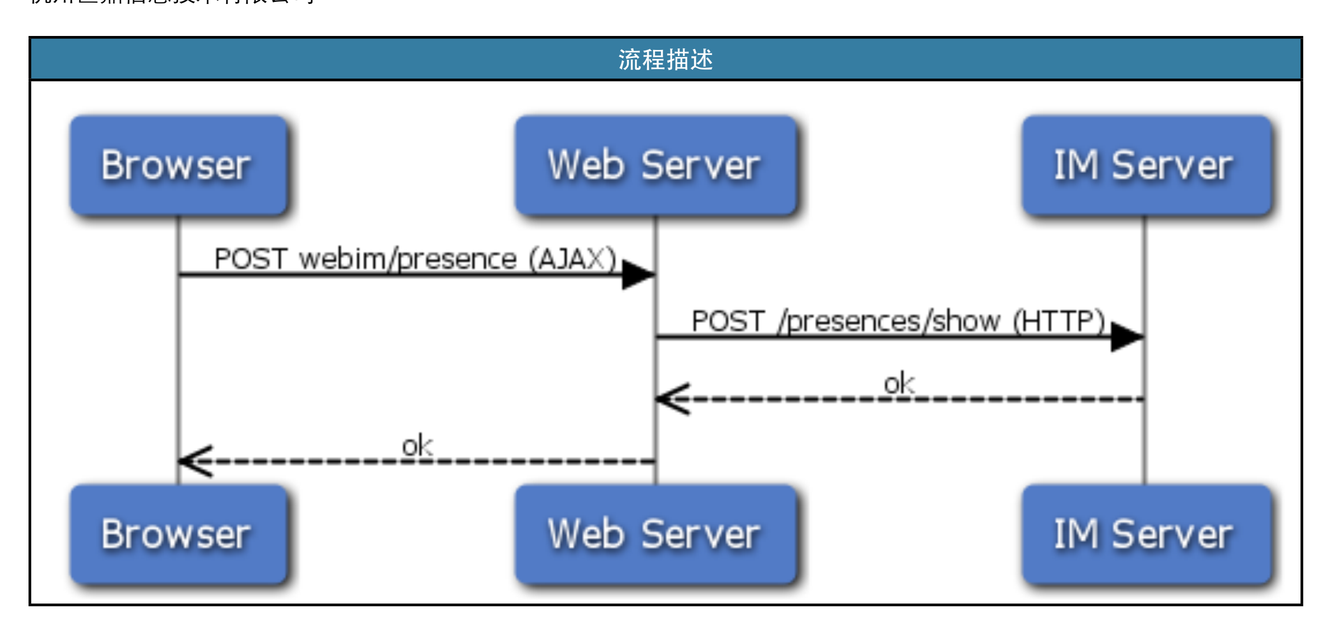
用户现场(Presence)状态变更流程

用户可通过“联系人”界面上方的下拉条，更改自身的现场(Presence)状态，操作界面如下：



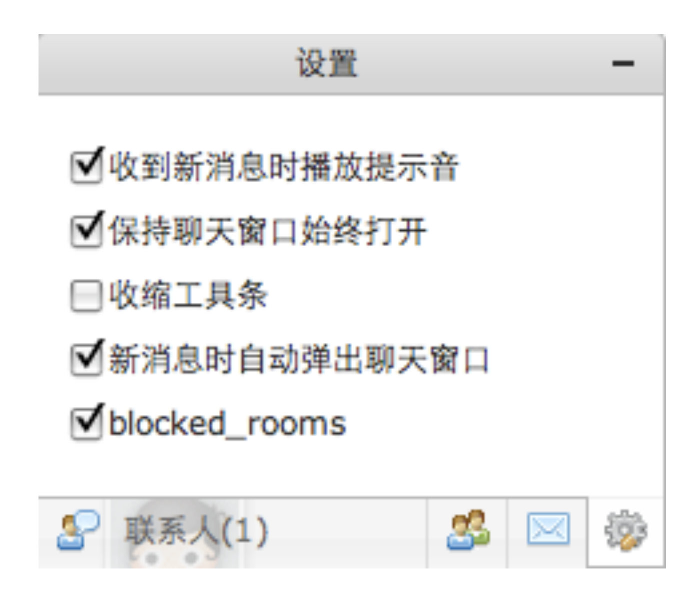
流程如下：

流程描述
Browser->Web Server: POST webim/presence (AJAX) Web Server->IM Server: POST /presences/show (HTTP) IM Server-->Web Server: ok Web Server-->Browser: ok
流程图



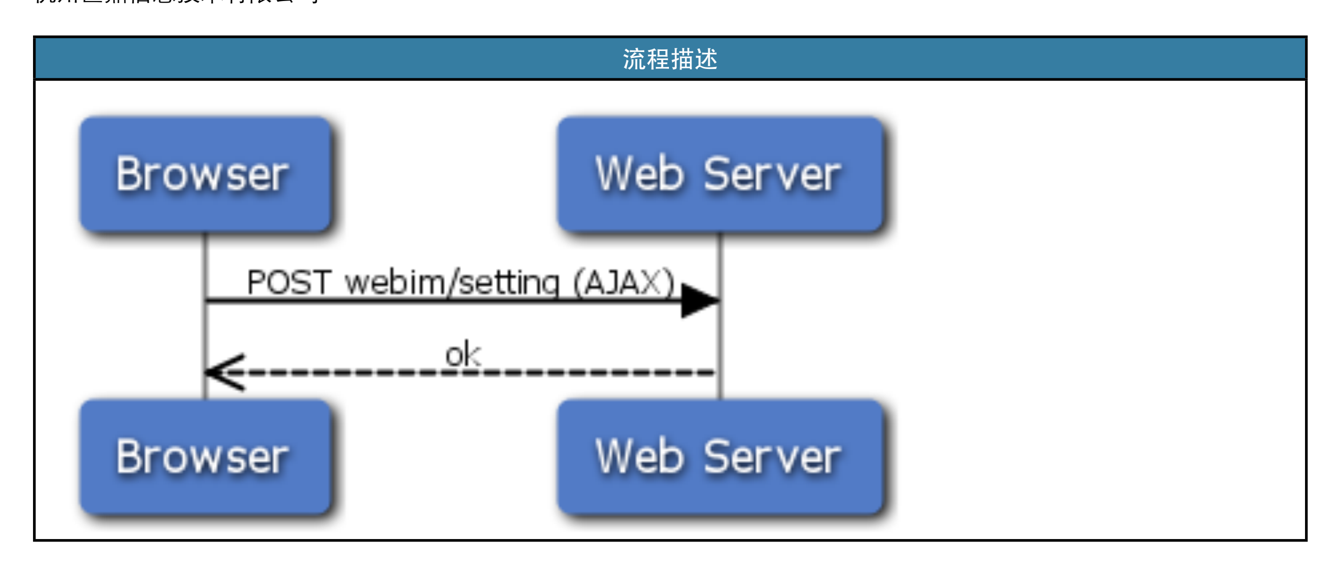
用户个人设置变更流程

用户可通过“设置”界面，更改个性化设置，操作界面如下：



流程如下：

流程描述
Browser->Web Server: POST webim/setting (AJAX) Web Server-->Browser: ok
流程图

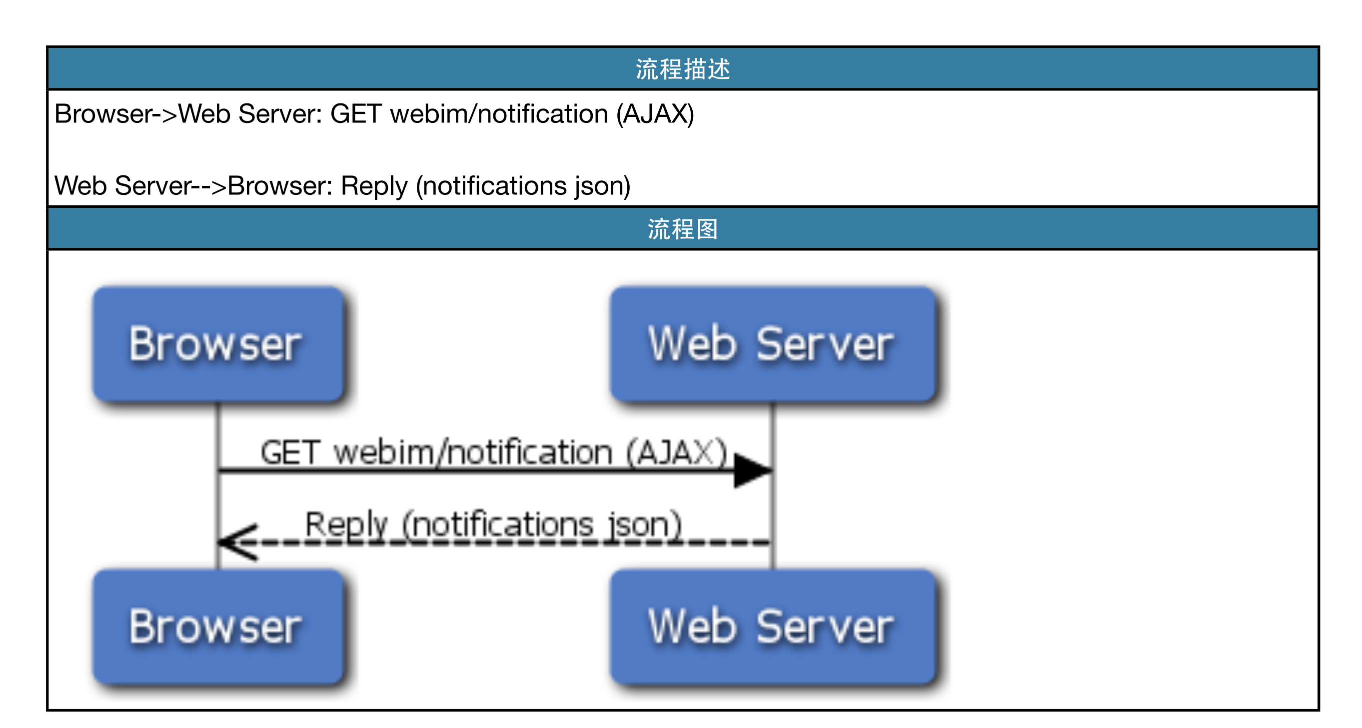


获取站内通知流程

用户可在“通知”界面，查看站内通知，操作界面如下：



流程如下：



流程描述
Browser->Web Server: GET webim/notification (AJAX) Web Server-->Browser: Reply (notifications json)
流程图

NextIM接口设计

客户浏览器->社区服务器接口

常用数据

链接信息：connection

参数名	必须	说明	例子
domain	TRUE	域名，社区服务器与消息服务器通信认证的域名；	www.uchome.com
ticket	TRUE	本次通信令牌，用于浏览器与消息服务器建立JSONP长连接	8633d182-b7fe-42a3-8466-1c4134cfebf2
server	TRUE	im服务器url	http://ucim.webim20.cn:8000

示例：

```
{
    "domain": "www.uchome.com",
    "ticket": "8633d182-b7fe-42a3-8466-0c4134cfebf2",
    "server": "http://www.webim20.cn:8000"
}
```

用户信息：userInfo

参数名	必须	说明	例子
id	TRUE	用户唯一ID	一般是username，而不是数据库的ID
nick	TRUE	用户昵称	
url	FALSE	用户个人页面链接	
pic_url	FALSE	用户头像图片链接	
default_pic_url	FALSE	用户头像默认地址	
show	FALSE	用户presence信息['available', 'away', 'chat', 'dnd', 'busy', 'invisible']	available
status	FALSE	用户状态消息	I'm not here right now.
status_time	FALSE	用户状态消息时间	10:55

示例：

{
  "id": 1,
  "nick": "Jack",
  "pic_url": "http://t1.qlogo.cn/mbloghead/c39e95b85b1b6bcd6f84/50",
  "default_pic_url": "http://mat1.gtimg.com/www/mb/images/head_50.jpg",
  "url": "space.php?uid=2",
  "status": "I'm free.",
  "show": "available",
  "status_time": "10:55"
}

联系人信息 :buddyInfo

参数名	必须	说明	例子
id	TRUE	用户唯一标示，一般是登陆账户	
nick	TRUE	用户昵称	
presence	TRUE	联系人是在线离线["online", "offline"]	
need_reload	FALSE	是否需要重载好友信息，返回信息不完整时使用，默认false	
group	FALSE	用户所属分组	friend
url	FALSE	用户个人页面链接	
pic_url	FALSE	用户头像图片链接	
default_pic_url	FALSE	用户默认头像（当用户头像加载失败时使用）	
show	FALSE	用户在线信息['available', 'away', 'chat', 'dnd', 'busy', 'invisible']	available
status	FALSE	用户状态消息	I'm not here right now.
status_time	FALSE	用户状态消息时间	10:55
history	FALSE	联系人和当前用户聊天记录，如果没有则会新建连接从webim/history读取	

示例：

{
  "id": 2,

"nick": "Susan",

"group": "friend",

"pic_url": "http://t1.qlogo.cn/mbloghead/a061e4ecb5b1ecd6ccee/50",

"default_pic_url": "http://mat1.gtimg.com/www/mb/images/head_50.jpg",

"url": "space.php?uid=2",

"presence": "online",

"status": "I'm buzy.",

"show": "buzy",

"status_time": "10:55",

"history": &history

}

群组成员信息：roomMemberInfo

参数名	说明	例子
id	用户唯一ID，一般是登陆账户	
nick	用户昵称	

示例：

{

"id": 1,

"nick": "Jack"

}

群组信息：roomInfo

参数名	必须	说明	例子
id	TRUE	群组id	1
nick	TRUE	群组名称	
url	FALSE	群组页面链接	
pic_url	FALSE	群组图片链接	
default_pic_url	FALSE	群组默认图片（当群组图片加载失败时使用）	
all_count	TRUE	所有群组成员总数	
count	TRUE	在线群组成员数	
blocked	FALSE	该用户是否屏蔽了次群组	FALSE

参数名	必须	说明	例子
members	FALSE	群组在线成员，如果没有会新建连接从webim/members读取	
history	FALSE	群组聊天记录，如果没有则会新建连接从webim/history读取	

示例：

```
{
    "id": "2",
    "nick": "Jack",
    "pic_url": "http://www.uchome.com/uc/ucenter/avatar.php?uid=2&size=small&type=virtual",
    "default_pic_url": "",
    "url": "group.php?uid=2",
    "all_count": 10,
    "count": 5,
    "blocked": false,
    "members": &members,
    "history": &history
}
```

历史记录信息：logItem

参数名	必须	说明	例子
from	TRUE	消息发送用户	
nick	TRUE	消息发送用户昵称	
to	TRUE	消息目的用户，一般为用户或者群组	
timestamp	TRUE	消息时间戳,时间为javascript时间，php中使用microtime(true)*1000	
type	TRUE	消息类型:unicast, multicast, broadcast	
body	TRUE	消息主体内容	
style	FALSE	消息style，CSS格式	

示例

```
{
        "type": "unicast",
```

"to": “alice”,

"from": “jack”,

"nick": "Jack",

"style": "color:#bbb;",

"body": "Hello.",

"timestamp": 1246883572400

}

联系人列表：buddies

[&buddyInfo]

群组列表：rooms

[&roomInfo]

用户上线：**POST /webim/online**

请求参数:

参数名	说明	例子
stranger_ids	当前网页显示的未知联系人，逗号分隔	13, 2, 5
buddy_ids	显示在tabs中的联系人列表，需要online后取得联系人信息和聊天记录	5, 8, 9
room_ids	显示在tabs中的群组列表，需要online后取得联系人信息和聊天记录	

返回参数:

参数名	说明	例子
server_time	服务器当前时间，解决本地时差，返回js时间戳。microtime(true)*1000	
connection	链接信息	&connection
user	用户信息	&userInfo
buddy_online_ids	在线好友列表，逗号分隔	5, 8, 9
buddies	根据请求参数中buddy_ids和离线消息取得联系人信息	&buddies
rooms	所有群组列表	&rooms

参数名	说明	例子
histories	根据请求时buddy_ids,room_ids,离线消息联系人取得联系人聊天历史记录	{susan: [&logItem]}
new_messages	未收到的离线消息	[&logItem]

发送消息：POST /webim/message

请求参数

&connection

参数名	说明	例子
offline	是否离线消息1为离线，0为在线	
to_name	接收者用户名	
to	接收者id	
type	消息类型:unicast, multicast, broadcast	
body	消息主体内容	
style	消息style，CSS格式	

返回参数

ok

发送状态：POST /webim/status

请求参数

&connection

参数名	说明	例子
to_name	接收者用户名	
to	接收者id	
show	状态显示内容	typing.....

返回参数

ok

设置参数：POST /webim/setting

请求参数

&connection

参数名	说明	例子
play_sound	是否播放声音	TRUE
buddy_sticky		
minimize_layout	缩小工具条	
msg_auto_pop	自动弹出消息	

返回参数

ok

用户离线：POST /webim/offline

请求参数

&connection

返回参数

ok

用户刷新页面：POST /webim/refresh

请求参数

&connection

返回参数

ok

获得联系人信息：GET /webim/buddies

请求参数：

ids:"2,3"

返回参数

[&buddyInfo]

获得群组信息：GET /webim/rooms

请求参数：

ids:"2,3"

返回参数

[&roomInfo]

加入群组：POST /webim/join

请求参数：

&connection

id:5

name:"room5"

返回参数

&roomInfo

离开群组：POST /webim/leave

请求参数：

&connection

id:5

name:"room5"

返回参数

ok

获得群组成员：GET /webim/members

请求参数：

&connection

ids:"5,6"

names:"room5,room6"

返回参数

```
{
    5:[&roomMemberInfo],
    6:[&roomMemberInfo]
}
```

获得聊天历史记录：GET /webim/histories

请求参数：

ids: "5,6"

type: unicast

返回参数：

```
{
    5:[&logItem],
    6:[&logItem]
}
```

清除聊天历史记录：POST /webim/clear_history

请求参数：

id: 5

返回参数：

ok

社区服务器->消息服务器接口

用户上线: POST /presences/online

请求参数:

参数名	说明	例子
domain	域名，社区服务器与消息服务器通信认证的域名；	
apikey	APIKEY，通信密钥	
name	用户名	
nick	用户昵称	
buddies	用户好友列表，逗号分割	a,b,c,d
rooms	用户所属群组列表，逗号分割	room1,room2,room3

正常返回为JSON格式对象，包括下述属性:

属性名	说明	例子
ticket	本次通信令牌，用于浏览器与消息服务器建立JSONP长连接	
buddies	用户在线好友列表，包括好友状态，JSON对象格式，见例子	{a: “available”, b: “dnd”, c: “hidden”}
roominfo	用户所属群组的当前会员总数，JSON对象格式，见例子	{“room1”: 10, “room2”: 20”}
clientnum	暂未用，向前兼容	

用户离线: POST /presences/offline

请求参数:

参数名	说明	例子
domain	域名，社区服务器与消息服务器通信认证的域名；	
apikey	APIKEY，通信密钥	
ticket	本次通信令牌	

正常返回: “ok”

错误返回:

错误码	错误信息	错误说明
404	no client found	找不到该用户
404	invalid ticket	通信令牌错误

用户现场变更: POST /presences/show

请求参数:

参数名	说明	例子
domain	域名，社区服务器与消息服务器通信认证的域名；	
apikey	APIKEY，通信密钥	
ticket	本次通信令牌	

参数名	说明	例子
nick	用户昵称	
show	现场Show信息	available', 'away', 'chat', 'dnd', 'invisible'
status	现场状态信息	Available, I'm busy

正常返回: “ok”

用户输入状态: POST /statuses

请求参数:

参数名	说明	例子
domain	域名，社区服务器与消息服务器通信认证的域名；	
apikey	APIKEY，通信密钥	
ticket	本次通信令牌	
to	发送目标好友名称	
nick	用户昵称	
show	状态展示信息	“xxx正在输入...”

正常返回: “ok”

用户发送消息: POST /messages

请求参数:

参数名	说明	例子
domain	域名，社区服务器与消息服务器通信认证的域名；	
apikey	APIKEY，通信密钥	
ticket	本次通信令牌	
to	发送目标好友名称	
nick	用户昵称	
body	消息主体	
style	消息style，CSS格式	
timestamp	消息时间戳	
type	消息类型: unicast, multicast,broadcast	

正常返回: “ok”

读取在线用户列表: GET /onlines

请求参数:

参数名	说明	例子
domain	域名，社区服务器与消息服务器通信认证的域名；	
apikey	APIKEY，通信密钥	

参数名	说明	例子
ticket	本次通信令牌	
names	要查询的在线用户列表，包括用户状态，JSON对象格式，见例子	{a: “available”, b: “dnd”, c: “hidden”}

正常返回逗号分割的在线用户名称文本，例如”a,b,c”。

读取群组用户列表: GET /room/members

请求参数:

参数名	说明	例子
domain	域名，社区服务器与消息服务器通信认证的域名;	
apikey	APIKEY，通信密钥	
ticket	本次通信令牌	
room	群组名称或ID	room1

正常返回该群组当前所有在线用户的JSON对象，格式如下:

{“room1”: [{“id”: “user1”, “nick”: “user1”}, {“id”: “user2”, “nick”: “user2”}]}

客户浏览器->消息服务器接口

JSONP长轮训读取消息: GET /packets

请求参数:

参数名	说明	例子
domain	域名，社区服务器与消息服务器通信认证的域名;	
ticket	本次通信令牌	
callback	JSONP回调函数名称	

返回JSONP执行脚本: callback(Object)，Object对象结构:

属性名	说明	例子
status	状态，一般为: ”ok”	
messages	用户接收到的消息列表，具体消息对象见下面消息对象说明	[Msg1, Msg2, ...]
presences	用户接收到的好友现场信息列表，具体现场对象见下面说明	[Presence1, Presence2, ...]
statuses	用户接收到的好友状态信息列表，具体状态对象见下面说明	[Status1, Status2, ...]

消息对象结构:

属性名	说明	例子
from	消息发送用户	
nick	消息发送用户昵称	
to	消息目的用户，一般为用户本身或者群组	
timestamp	消息时间戳	
type	消息类型:unicast, multicast, broadcast	
body	消息主体内容	
style	消息style，CSS格式	

状态对象结构:

属性名	说明	例子
from	状态发送用户	
nick	消息发送用户昵称	
to	消息目的用户	
show	状态信息内容	“xxx正在输入...”

现场对象结构:

属性名	说明	例子
from	状态发送用户	
nick	消息发送用户昵称	
type	现场类型:online, offline, show	
show	现场展现信息	available
status	现场状态信息	Available，Busy...

NextIM集群设计

NextIM的消息服务器本身基于Erlang设计，支持大规模集群。当社区站点并发用户规模超过50K以上，需要集群部署时，唯一需要做的工作是在社区服务器在转发消息之前，按用户名或ID进行用户切割。具体切割方案，例如:

- range切割，按用户id大小切割，比如1～10000分发到消息服务器1，10000～20000分发到服务器2
- hash切割，根据用户名称进行hash计算，然后分发到对应消息服务器

NextIM公司介绍

杭州巨鼎

杭州巨鼎(WEBIM20.CN)是以技术创新为导向的WEB即时通信产品与服务供应商，我们长期专注于WebIM产品的研发设计，针对不同类型的网站推出多种WebIM即时通信解决方案。 我们坚信企业“专业、专注、专长”的价值，坚持不懈的开发和完善以Erlang语言为核心的WebIM产品。我们是国内最早基于Erlang语言设计开发大规模并发WEBIM即时通信产品的公司，广泛用于在线客服、社区网站、论坛网站、商城网站。

联系我们

手机：(+86).18958095588

邮箱：ery.lee@gmail.com

电话：(+86)-0571-89719990-808

传真:(+86)-0571-89719991

地址：杭州市万塘路69号华星科技苑C座3层